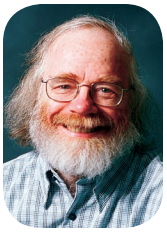
Security by Checklist

We've all seen the checklists with suggestions for how to secure your system. Many of us have even written them. (I have.) The problem is that security is more complicated than that, and checklists—especially if followed slavishly or enforced without thought—can make matters worse.



STEVE BELLOVIN
Columbia University

Passwords are one case in point. Users, we're told, should pick unguessable passwords and never write them down. System developers should ensure that passwords are changed frequently; furthermore, they should never store them in the clear. These rules are simple and obvious—but often (though not always), they're wrong.

The problem with unguessable passwords, especially if they're changed frequently, is that they're too often forgotten. At that point, backup methods come into play—your mother's maiden name, your favorite color, email reminders and so on. These are generally *less* secure. Besides, the most common way passwords are compromised today is via keystroke loggers, not guesses—and keystroke loggers don't care if your password is a modified version of the second letters of the fourth verse of the Klingon national anthem.

One-way password hashing seems like an obvious solution; after all, it protects the password file against theft. However, doing that rules out reminder notices (very important for low-value Web passwords); more important, you can't use a hashed password for challenge-response authentication, negotiating a shared cryptographic session key, or for other such things.

The hashed password example illustrates the real point: there is a trade-off not just between security and functionality, but between different aspects of security. Which is more important, protecting against a stolen password file or being able to generate a shared session key? A checklist can't answer that question, however well-intentioned.

Other checklist entries are more or less harmless, if you don't count the wasted effort. We're told "always use a firewall" but why? Firewalls can be very helpful, in the right environment and with the proper configuration, but in other situations, they add no security— it's rarely useful to protect a public Web server with a firewall, for example. Port 80 is the most dangerous port on the server, but it's the one port that *must* be allowed through.

Other times, firewalls can cause a false sense of security. Fundamentally, firewalls are policy enforcement points, so what matters, more than anything, is the policy they enforce. If the policy is too lax, the firewall does no good whatsoever. Complex policies—that is, the kind of policies necessary in many large organizations—are error-prone.[1] What's the benefit of such a firewall here?

Even obvious correct advice on checklists can be bad. One I consulted when writing this column said "DO use the latest version of Eric Allman's sendmail." (It gives the same advice for many other packages.) But installing a new version isn't always simple; a few sentences later, the very same document notes that "other sendmail configuration files may not be compatible with the latest version of sendmail." Furthermore, patches are often incompatible with vital applications. Which is worse, to risk a penetration or to be off the air completely because your applications don't work? The answer, of course, is "it depends."

That's the real danger with checklists: they try to substitute simple adherence to rules for thought. A checklist is fine as a reminder, but it should *never* be used as a blind substitute for analysis. □

Reference

1. A. Wool, "A Quantitative Study of Firewall Configuration Errors," *Computer*, vol. 37, no. 6, 2004, pp. 62–67.

Steve Bellovin *is a professor of computer science at Columbia University. He has a BA from Columbia University and an MS and PhD from the University of North Carolina at Chapel Hill. Bellovin helped create netnews, or usenet news, and is coauthor of* Firewalls and Internet Security *(Addison-Wesley, 2003). Contact him via www.cs.columbia.edu/~smb.*

40-7993/08/$25.00 © 2008 IEEE ■ IEEE SECURITY & PRIVACY ■ MARCH/APRIL 2008